UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/608,040	06/30/2003	Hajime Ogawa	2003_0866A	3923

513 7590 11/28/2008

WENDEROTH, LIND & PONACK, L.L.P.
2033 K STREET N. W.
SUITE 800
WASHINGTON, DC 20006-1021

EXAMINER	
WEI, ZHENG	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
11/28/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/608,040	OGAWA ET AL.	
	Examiner	Art Unit	
	ZHENG WEI	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *29 July 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *19-22,30,32,33,42,43,48-51 and 55-67* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *19-22,30,32,33,42,43,48-51 and 55-67* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. This office action is in response to the amendment filed on 07/29/2008.

2. Claims 1-5, 8-13, 35, 36, 40, 52-54 have been canceled.

3. Claims 19-22, 30, 32, 33, 42, 43 and 48-51 have been amended.

4. Claims 55-67 have been added

5. Claims 19-22, 30, 33, 42, 43, 48-51 and 55-67 remain pending and have been examined.

Response to Arguments

6. Applicant's arguments filed on 07/29/2008, in particular on pages 23-33, have been fully considered but they are not persuasive. For example:

- At page 24, first paragraph, the Applicants submit that Stallman does not disclose or suggest the features about the optimization unit allocates all the array data of the specific type declared in the source program in the memory region so that its head address matches the alignment. In particular, the Applicants points out at second paragraph that the Stallman does not disclose or in any way suggest that an optimization unit allocates all the array data of the specific type declared in the source program in the memory region so that its head address matches the alignment, as recited in Claim 30.

However, the Examiner's position is that claim recites term "array data" which can be reasonable interpreted as data in the array and each array has to be

declared as a specific data type before it can be used [emphasis added].

Therefore, the recited limitation "allocating all the array data of the specific type declared in the source program in the memory region so that its head address matches the alignment" is just allocating the specific type of the array in the memory according its declaration. When the specific type of array is declared/allocated in the memory, all the array data and its address in that array is also decided accordingly. As Stallman disclosed the example about aligning the array with specific data type "short" using a directive "aligned", all the array data in said short type of array is also aligned (see for example, p.178) which reads the claim limitation.

- At page 24, forth paragraph, the Applicants submit that Stallman does not disclose or suggest the recited feature of claim 32 that the directive acquisition unit detects a designation of alignment of data that a pointer variable of argument shown by name of specific variable indicates in the source program, wherein the optimization unit performs the optimization assuming that the data that is an object of designation detected by the directive acquisition unit is allocated in the memory region by the designated alignment.

However, the Examiner respectfully disagrees.

It should be noted that pointer and array are the same mechanism for accessing the data in the memory by merely using two different terms (either pointer variable or the name of the specific array variable). As the Applicants

recited “the directive acquisition unit detects a designation of alignment of data that a pointer variable of argument shown by the name of a specific variable indicates in the source program”, it merely indicates "a designation of alignment of data” and “a pointer variable of argument shown by the name of a specific variable indicates in the source program”, but does not define any relation between “designation of alignment of data” and “pointer variable”/the name of a specific variable indicates in the source program”. Moreover, as Stallman disclosed an example about aligning array data “short array[3] __attribute__ ((aligned)):” (see for example, p.177-178), it shows that the short type of array with the size 3, which has 3 variables array[0], array[1] and array[2], each array variable is a pointer variable which is shown by the name of the array with a index to indicate the memory address of specified variable and thus aligning a pointer variable (data) is the same as “alignment of the array data”. Therefore, the alignment feature as Stallman disclosed above for the array variable also discloses the same aligning for the pointer variable in the memory.

At page 25, last paragraph, the Applicants submit that the recited limitation in claim 33 regarding "a local pointer variable" is not taught by Stallman. The Examiner respectfully disagrees by the same reason as addressed in claim 32 above about the "pointer variable".

- At pages 27-28, the Applicants request clarification regarding the reason for rejection to claims 19 and 48. The Examiner’s position is that the claim

language does not provide further limitation about using directive for optimization by loop unrolling, e.g. the recited claim language "wherein the directive acquisition unit detects designation of the number of iterations of specific loop processing in the source program". However, nowhere in the claims define any relationship between the "directive" and "designation of the number of iterations of specific loop" [emphasis added]. Thus there is no further limitation about said directive has to include and/or define said designation of number of iteration of specific loop as recited in the claims and can be considered merely general descriptions e.g. could be interpreted as any loop number in the source code with required number of time to iterate and actual running the number of iteration of the loop. Moreover, the recited "designated number of iterations" in claim "wherein the optimization of loop processing that is an object of the designation detected by the directive acquisition unit based on the designated number of iterations" and "wherein the designation of the number of the iterations is the minimum number by which the loop processing is iterated" and "wherein the optimization unit restrains generation of an escape code that is needed in the case of the number of the iterations being 0 when the minimum number is 1 or more" is not necessary to be considered as the directive [emphasis added]. It could be "designation of the number of iterations" of the loop itself in source code being detected. Therefore as Stallman disclosed directive/compile option feature, PGI discloses information about using directive (pragmas) to perform "unroll"

and "nounroll" feature and Geva discloses detecting loop iteration, said recited limitation is obvious in view of the combination of these reference as addressed above.

- In regarding to claims 20, 49, 21, 50, 22 and 51, for the same reason as addressed above, the "directive" does not define any relationship with the designation of the number of iteration and thus detected "designation of the number of iteration" does not further limit the definition of the "directive" and can be considered merely general descriptions e.g. could be interpreted as any loop number in the source code with required number of time to iterate and actual running the number of iteration of the loop. Therefore, different specified number of iterations e.g. even number, odd number as recited in the above claims are just the implementation to deal with different predefined condition.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 30, 32, 42, 33 and 43 are rejected under 35 U.S.C. 102(b) as being anticipated by Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1)

Claim 30:

Stallman discloses a computer-implemented compiler apparatus having instructions stored thereon for causing a computer to translate a source program into a machine language program, said compiler apparatus comprising:

- a directive acquisition unit operable to acquire a directive for optimizing a machine language program to be generated (see for example, p.9, "Optimization options"); and
- an optimization unit operable to perform optimization by generating a sequence of machine language instructions following the acquired directive (see for example, see p.49, section 3.10, "Options that Control Optimization"),
- wherein the optimization unit performs optimization by allocating data in a memory region following a directive when the optimization unit acquires the directive on alignment of the array data to be allocated in a memory region (see for example, p.177, section 5.33 Specifying Attributes of Variables. "aligned (alignment)" and related description, also see example, p.178, line 7 and line 19, "short array[3] __attribute__ ((aligned)):" and related description)

- wherein the directive acquisition unit acquires a directive for alignment of array data of a special type together with a directive for translating the source program (see for example, p.10, line 3, "-O –O0 –O1 –O2 -O3 –Os" and related text), and
- the optimization unit allocates all the array data of the special type declared in the source program in the memory region so that its head address matches the alignment (see for example, p.177, section 5.33 Specifying Attributes of Variables. "aligned (alignment)" and related description, also see example, p.178, line 1-19, "short array[3] __attribute__ ((aligned)):" and related description).

Claim 32:

Stallman discloses a computer-implemented compiler apparatus having instructions stored thereon for causing a computer to translate a source program into a machine language program, said compiler apparatus comprising:

- a directive acquisition unit operable to acquire a directive for optimizing a machine language program to be generated (see for example, p.9, "Optimization options"); and
- an optimization unit operable to perform optimization by generating a sequence of machine language instructions following the acquired directive (see for example, see p.49, section 3.10, "Options that Control Optimization"),

- wherein the optimization unit performs optimization by allocating data in a memory region following a directive when the optimization unit acquires the directive on alignment of the array data to be allocated in a memory region (see for example, p.177, section 5.33 Specifying Attributes of Variables. "aligned (alignment)" and related description, also see example, p.178, line 7 and line 19, "short array[3] __attribute__ ((aligned)):" and related description)
- wherein the directive acquisition unit detects designation of alignment of data that a pointer variable of argument shown by the name of a specific variable indicates in the source program (see for example, p.182, lines 16-20, "If you declare or use arrays of variable of an efficiently-aligned type…") and
- the optimization unit performs the optimization assuming that the data that is an object of designation detected by the directive acquisition unit is allocated in the memory region by the designated alignment (see for example, p.182, lines 16-20, "the compiler generates for these pointer arithmetic operations…").

Claim 42:

Stallman also discloses the compiler apparatus according to claim 32, wherein the optimization unit generates a pair instruction for transferring two or more kinds of data at the same time regarding a memory access instruction for

accessing the data to be allocated in the memory region (see for example, p.181, lines 26-29, “use the ldd and std (doubleword load and store) instructions”).

Claim 33:

Stallman discloses a computer-implemented compiler apparatus having instructions stored thereon for causing a computer to translate a source program into a machine language program, said compiler apparatus comprising:

- a directive acquisition unit operable to acquire a directive for optimizing a machine language program to be generated (see for example, p.9, “Optimization options”); and
- an optimization unit operable to perform optimization by generating a sequence of machine language instructions following the acquired directive (see for example, see p.49, section 3.10, “Options that Control Optimization”),
- wherein the optimization unit performs optimization by allocating data in a memory region following a directive when the optimization unit acquires the directive on alignment of the array data to be allocated in a memory region (see for example, p.177, section 5.33 Specifying Attributes of Variables. “aligned (alignment)” and related description, also see example, p.178, line 7 and line 19, “short array[3] __attribute__ ((aligned)):” and related description)

- wherein the directive acquisition unit detects designation of alignment of data that a local pointer variable shown by the name of a specific variable indicates in the source program (see for example, p.182, lines 16-20, "If you declare or use arrays of variable of an efficiently-aligned type…") and
- the optimization unit performs the optimization assuming that the data that is an object of designation detected by the directive acquisition unit is allocated in the memory region by the designated alignment (see for example, p.182, lines 16-20, "the compiler generates for these pointer arithmetic operations…").

Claim 43:

Stallman also discloses the compiler apparatus according to claim 33, wherein the optimization unit generates a pair instruction for transferring two or more kinds of data at the same time regarding a memory access instruction for accessing the data to be allocated in the memory region (see for example, p.181, lines 26-29, "use the ldd and std (doubleword load and store) instructions").

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 19-22 and 48-51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Stallman (Richard M. Stallman, Using and Porting the GNU Compiler Collection for GCC 3.1) in view of PGI (PGI Workstation User's Guide-9 Optimization Directive and Pragmas) and further in view of Geva (Robert Y. Geva, US 6,539,541)

Claim 19:

Stallman discloses a computer-implemented compiler apparatus having instructions stored thereon for causing a computer to translate a source program into a machine language program, said compiler apparatus comprising:

- a directive acquisition unit operable to acquire a directive for optimizing a machine language program to be generated (see for example, p.9, "Optimization options"); and
- an optimization unit operable to perform optimization by generating a sequence of machine language instructions following an acquired directive (see for example, see p.49, section 3.10, "Options that Control Optimization"),
- loop unrolling option(see for example, see p.49, section 3.10, "Options that Control Optimization", and also see p.55, lines 38-42, "-funroll-loops" and related description).

but does not explicitly disclose wherein the optimization unit performs optimization by loop unrolling following a directive when the directive acquisition unit acquires the directive on the optimization by loop unrolling.

However, PGI in the same analogous art of compiler software optimization using directive discloses adding pragmas to C and C++ specific to perform or not perform loop unrolling (see for example, section 9.4 Adding Pragmas to C and C++, Table 9-2 C/C++ Pragma Summary, "unroll" and "nounroll" and related description). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to also add directive to gcc compiler to selectively perform or not perform loop unrolling optimization. One would have been motivated to do so selectively perform or not perform loop unrolling optimization to specific loop instead of all software programs.

But neither of them discloses detecting a directive of designation of the number of iteration. However, Geva in the same analogous art of loop unrolling discloses the number of iterations defined by the directive (see for example, col.9, lines 37-38, "where the value of loop iterations is read by the program from an input file"). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to specify the number of iterations of specific loop processing in the source program. One would have been motivated to do so to ensure the number of iterations can be determined at compile time and further guarantee the loop unrolling can be performed as suggest by Geva (see for example, col.9, lines 31-42)

Geva further discloses: the optimization unit restrains generation of an escape code that is needed in the case of the number of the iterations being 0 when the minimum number is 1 or more (see for example, col.10, lines 9-10, "When the unrolled loop is a counted loop, there is no need to test for the exit condition inside the unrolled body.").

Claim 20:

As per Stallman, PGI and Geva disclosed above are incorporated, Geva further disclose the optimization unit performs the optimization by loop unrolling when the minimum number is equivalent to or more than the number of development by the loop unrolling (see for example, col.10, lines 6-9, "One such optimization may be loop unrolling where a loop is unrolled 'n' times, such that 'n-1' additional copies of the loop body are made").

Claims 21 and 22:

As per Stallman, PGI and Geva disclosed above are incorporated, but neither of them explicitly discloses the number of iteration is even/odd number. However, Geva in the same analogous art of loop unrolling discloses the number of iterations defined by the directive (see for example, col.9, lines 37-38, "where the value of loop iterations is read by the program from an input file"). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to specify the number of iterations of specific loop

processing in the source program. One would have been motivated to do so to ensure the number (even/odd number) of iterations can be determined at compile time and further guarantee the loop unrolling can be performed as suggest by Geva (see for example, col.9, lines 31-42).

Claims 48-51:

Claims 48-51 are other version of device claims performing the similar claimed method as in claims 19-22 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly a computer system would need to run and/or practice such function steps disclosed by reference above. Thus, they also would have been obvious.

Claims 55-67:

Claims 55-67 are computer program product version claims, wherein all claimed limitation functions have been addressed in claims 19-22, 30, 32-33, 42-43 and 48-51 above respectively. It is well known in the computer art that such computer software including compiler apparatus can be practiced and /or stored on a computer readable recording media. Thus, they also would have been obvious in view of reference teachings above.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Applicant's arguments with respect to claims rejection have been considered but are persuasive. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571) 270-1059 and Fax number is (571) 270-2059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The

fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Z. W./
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192